1. INTRODUCTION

1.1 Project Overview

Snack Squad is an Android application that simplifies snack ordering for events such as movie nights or casual gatherings. Users can register, browse a catalog of snacks with images and prices, customize options (flavor, size, packaging), add items to a cart, choose delivery methods, pay securely, track the order in real time, and leave feedback.

1.2 Purpose

The project aims to offer a lightweight, user-friendly mobile solution that makes snack purchasing quick, customizable, and transparent—especially for students and young professionals who value time and convenience.

2. LITERATURE SURVEY

2.1 Existing Problem

General food-delivery apps focus on full meals, provide limited snack customization, and often have cluttered workflows. Users who just need snacks encounter slow checkouts and no dedicated tracking for small orders.

2.2 References

- Google Material Design Documentation
- “Mobile Commerce Trends in Food Delivery” – ACM Digital Library
- Firebase Docs: Realtime Database & Authentication

2.3 Problem Statement Definition

Event hosts and snack lovers lack a specialized, affordable mobile platform that lets them quickly select, customize, and track snack deliveries without the complexity

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Prompt	Details
Who are we empathizing with?	Students, young professionals hosting gatherings
What do they need to do?	Order snacks fast, customize flavors, ensure on-time delivery
What do they see?	Generic meal apps, complicated UIs, unclear delivery ETAs
What do they say?	“I just want snacks, not a full meal.”
What do they do?	Visit multiple stores or apps, manually track orders
What do they hear?	Friends complaining about late snacks, social posts on quick deliveries
Think & feel?	Excited for event but anxious about preparation time
Pains	Time-consuming shopping, limited customization, uncertain delivery
Gains	One-stop snack app, real-time tracking, flexible payment

3.2 Ideation & Brainstorming

Brainstormed features included:

- Dedicated snack catalog with images
- Cart with live total updates
- Multi-payment (Card, UPI, COD)
- Order tracking timeline
- Feedback & ratings post-delivery

Prioritized Idea: Build a Kotlin-based Android app with local data storage (SharedPreferences) that supports snack customization, secure payment validation, order tracking, and feedback.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

- User registration/login
- View snack catalog and images
- Add, remove, or update cart items
- Checkout with delivery options
- Multi-payment validation
- Order tracking UI
- Feedback and rating submission

4.2 Non-Functional Requirements

- Response time < 2 s for catalog and cart updates
- Secure storage of user credentials
- Modular Kotlin codebase
- Works offline for browsing (cached data)

5. PROJECT DESIGN

5.1 Data Flow Diagram & User Stories

Sample User Story:
As a user, I want to select snacks and see the total price instantly so that I can decide quickly.

DFD (high-level):
User → Snack Catalog → Cart Manager ↔ Payment Module → Order Status → Feedback

5.2 Solution Architecture

- **UI Layer:** XML layouts + Material Components
- **Domain Layer:** Kotlin classes (Snack, CartItem, CartManager)
- **Data Layer:** SharedPreferences (local) / optional Firebase (cloud)

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Stack

- Android SDK 35, Kotlin 1.9
- AndroidX, Material Components
- Gradle build system

6.2 Sprint Planning & Estimation

Week	Focus	Deliverables
1	Design & Setup	UI mock-ups, project skeleton
2	Core Features	Catalog, Cart logic
3	Payment & Tracking	Payment validation, tracking screen
4	Feedback & Polish	Feedback form, bug fixes, docs

6.3 Sprint Delivery Schedule

Sprint 1 → Project skeleton and catalog sample data
Sprint 2 → Cart and checkout flow
Sprint 3 → Payment + order tracking
Sprint 4 → Feedback, icon, final APK

7. CODING & SOLUTIONING

Feature	File(s)	Brief Description
Catalog Display	SnackAdapter.kt, item_snack.xml	RecyclerView adapter and layout for snack list
Cart Management	CartManager.kt, CartActivity.kt	Handles add/remove, quantity, total price
Payment Screen	PaymentActivity.kt, activity_payment.xml	Multi-payment validation (Card/UPI/COD)
Order Tracking	OrderTrackingActivity.kt, activity_order_tracking.xml	Simulates status updates
Feedback	PaymentSuccessActivity.kt, activity_payment_success.xml	Rating bar and feedback form
Data Persistence	SharedPreferences	Stores user session and settings

8. PERFORMANCE TESTING

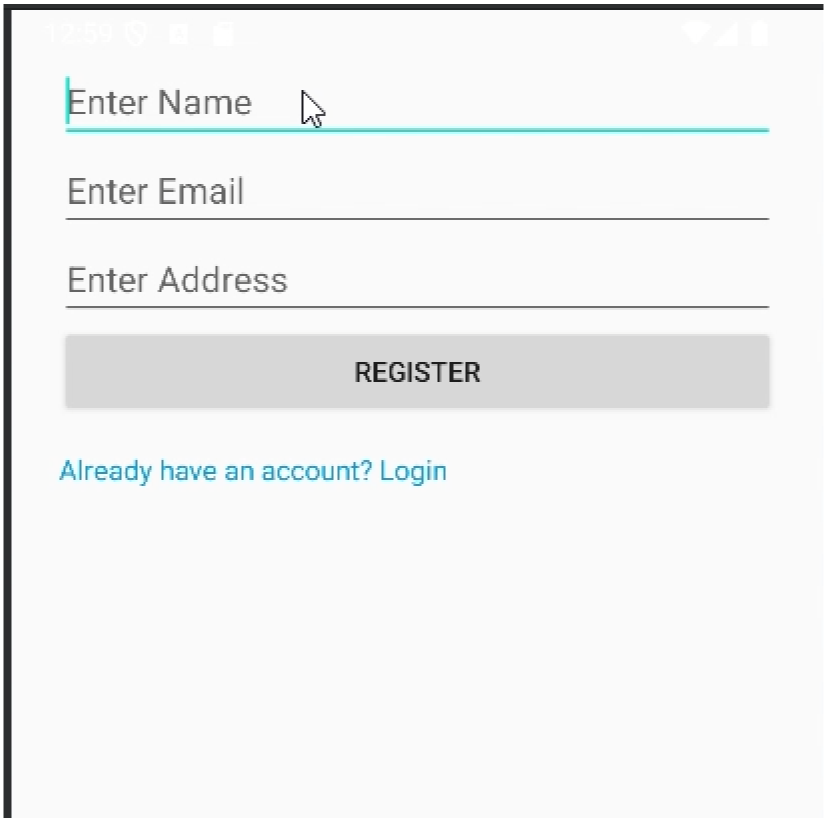
8.1 Performance Metrics

- Average screen-load time: < 1.5 s
- Payment validation latency: < 500 ms
- Memory usage: under 150 MB on emulator

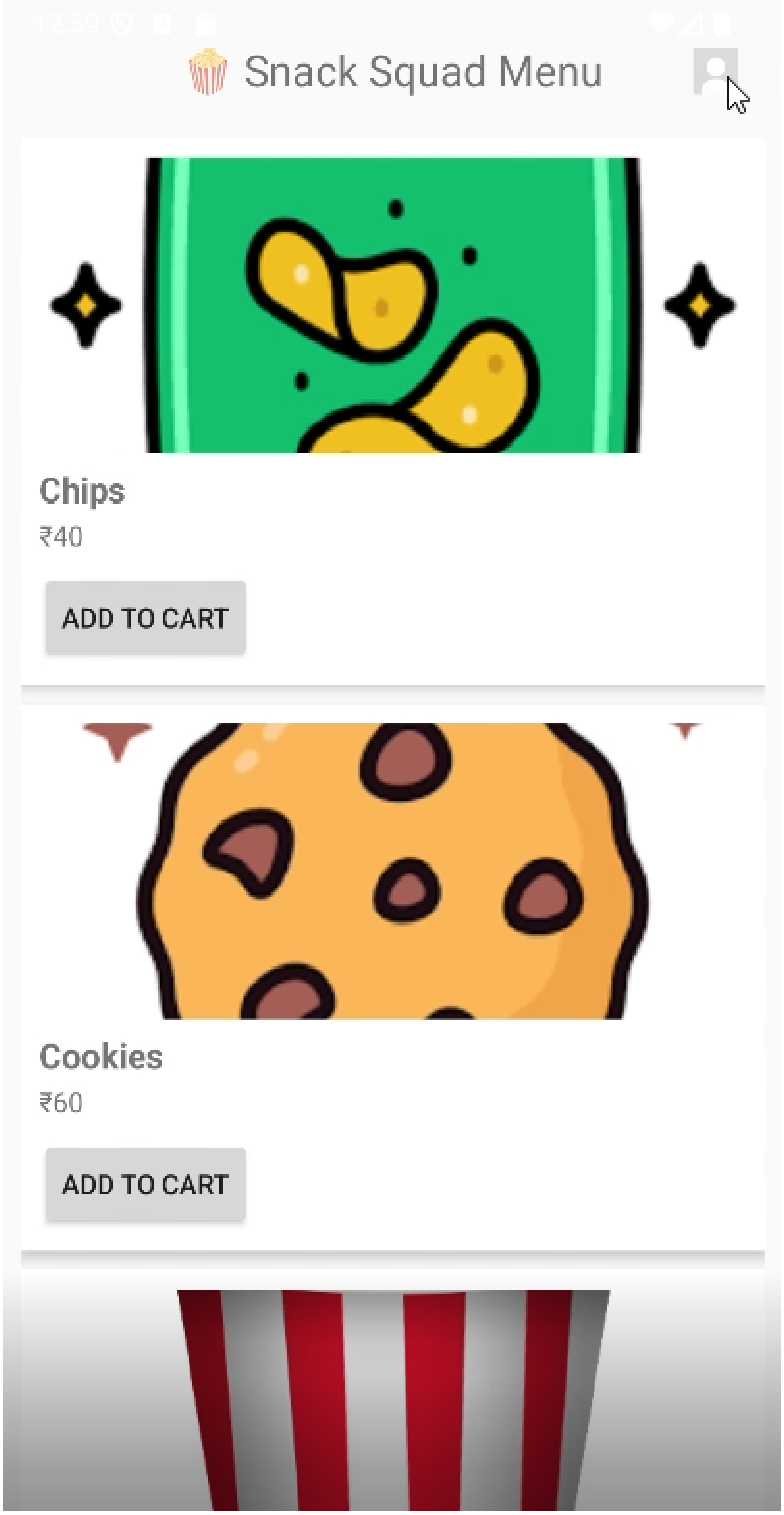
9. RESULTS

9.1 Output Screenshots

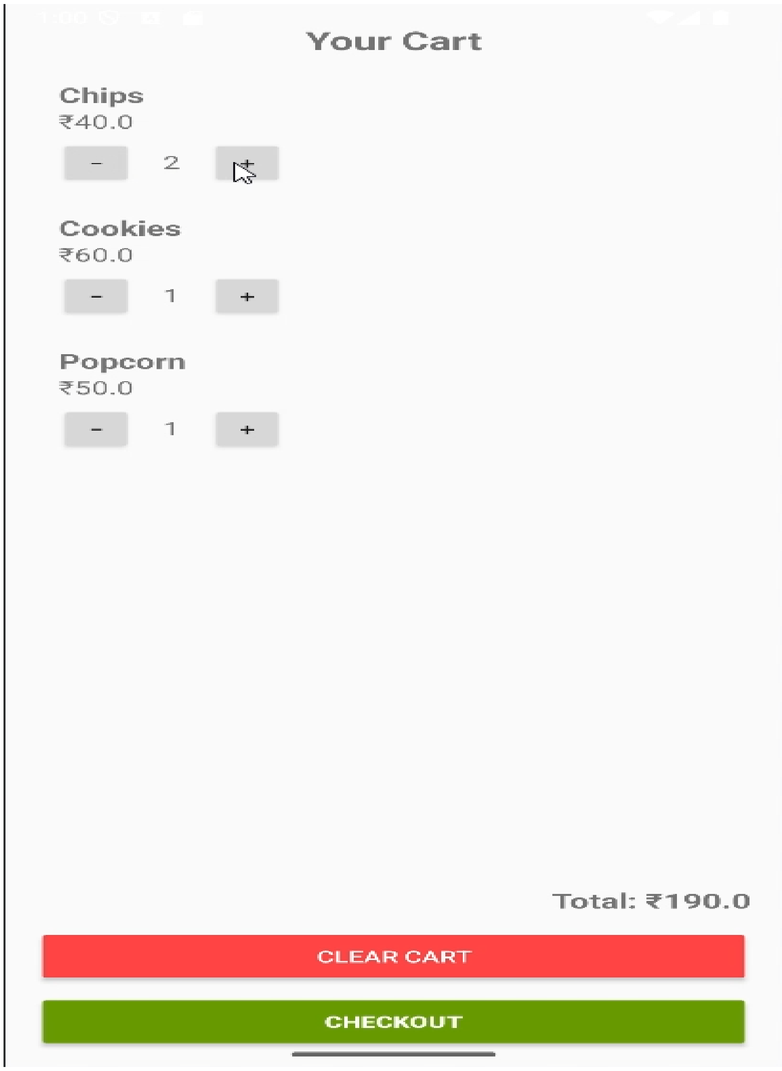
- Registration success –



- Snack catalog display



- Cart update with totals



Your Cart

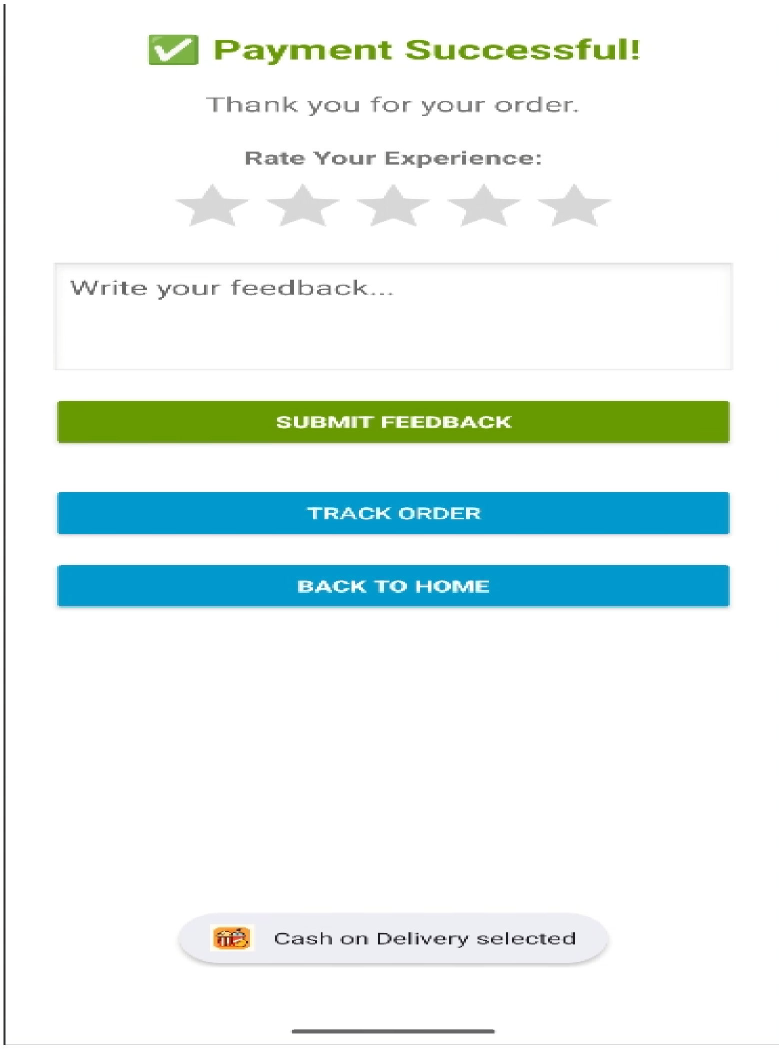
Chips
₹40.0
2

Cookies
₹60.0
1

Popcorn
₹50.0
1

Total: ₹190.0

- Payment success



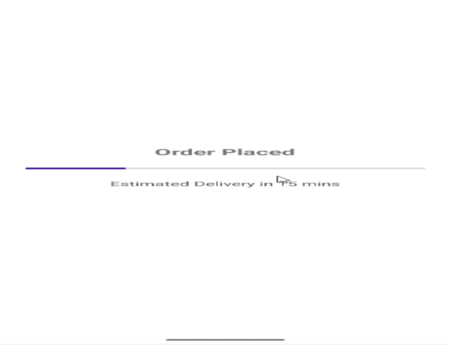
✅ Payment Successful!

Thank you for your order.

Rate Your Experience:

Cash on Delivery selected

- Order tracking timeline



Order Placed

Estimated Delivery in 15 mins

10. ADVANTAGES & DISADVANTAGES

Advantages

- Dedicated snack focus, minimal clutter
- Quick checkout with live totals
- Works offline for browsing cached catalog

Disadvantages

- Local-only data; no real vendor integration
- Payment module is simulated, not live gateway
- Basic tracking (no GPS integration)

11. CONCLUSION

Snack Squad demonstrates a compact yet complete mobile workflow for snack ordering—from registration to feedback. It shows how Kotlin and AndroidX can deliver a smooth user experience with essential e-commerce features, providing a foundation for future expansion.

12. FUTURE SCOPE

- Integrate Firebase for realtime vendor data
- Add push notifications for delivery status
- Implement live payment gateways (Razorpay, Google Pay)
- Introduce loyalty points and discount coupons

13. APPENDIX

- Source Code: Available in GitHub Repository
- GitHub Link: **https://github.com/YOUR_USERNAME/Snack-Squad-App**